DETECTION OF IMPERSONATION IN ONLINE EXAMINATIONS

A Mini Project Report

submitted by

ANJANA MS(MES20MCA-2009)

to

the APJ Abdul Kalam Technological University
in partial fulfillment of the requirements for the award of the Degree

of

Master of Computer Applications

Department of Computer Applications

MES College of Engineering
Kuttippuram, Malappuram - 679 582

February 2022

DECLARATION

I undersigned hereby declare that the project report **Detection of impersonation in online examinations**, submitted for partial fulfillment of the requirements for the award of degree of Master of Computer Applications of the APJ Abdul Kalam Technological University, Kerala, is a bona fide work done by me under supervision of Mr. Balachandran K P, Associate Professor, Department of Computer Applications. This submission represents my ideas in my own words and where ideas or words of others have been included, I have adequately and accurately cited and referenced the original sources. I also declare that I have adhered to ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of any degree, diploma or similar title of any other University.

Place:KUTTIPPURAM

Date:25/02/2022

ANJANA MS[MES20MCA-2009

DEPARTMENT OF COMPUTER APPLICATIONS
MES COLLEGE OF ENGINEERING, KUTTIPPURAM

CERTIFICATE

This is to certify that the report entitled **Detection of impersonation in online examinations** is a bonafide record of the Mini Project work carried out by **ANJANA MS(MES20MCA-2009)** submitted to the APJ Abdul Kalam Technological University, in partial fulfillment of the requirements for the award of the Master of Computer Applications, under my guidance and supervision. This report in any form has not been submitted to any other University or Institution for any purpose.

Internal Supervisor(s) External Supervisor(s)

Head Of The Department

Acknowledgements

My endeavor stands incomplete without dedicating my gratitude to a few people who have contributed towards the successful completion of my project.I pay gratitude to the Almighty for his invisible help and blessing for the fulfillment of this work.At the outset i express my heartful thanks to our guide Associate Professor Balachandran K P for his valuable guidance and supervision. I take this opportunity to express my profound gratitude to Mrs. Priya JD, as my project coordinator for her valuable support, timely advise and strict schedules to complete my project. I also grateful to all my teaching and non-teaching staff for their encouragement, guidance and whole-hearted support. Last but not least, I gratefully indebted to my family and friends, who gave me a precious help in doing my project.

ANJANA MS(MES20MCA-2009)

Abstract

Online Examination faces many security threats. The threats are of two types, intrusive and non-intrusive. Non-intrusive is the prominent threat faced in remote online examinations. Collusion is a type of nonintrusive threat, where a student invites a third party for impersonation or assistance during the examination. Collusion is comparatively difficult to detect and prevent because of the lack of efficiency and integrity on the part of the external invigilator. Impersonation is a significant problem in online examinations. An efficient invigilation mechanism is the need of the hour to ensure the standard of examination and to maintain the authentic conduct of the examination. Hence, i propose dynamic face authentication using the Viola-Jones algorithm and SVM to check the integrity of the candidate in the beginning of the examination.This approach aims in increasing the credibility of online exams and the exams can be taken from any convenient location. In recent days, the candidate appearing for an online examination is authenticated by carrying out manual verification of the candidate's credentials by the examiner. Conducting an automated face authentication will check the identity of the user when starting the exam. For this type of authentication, we use facial recognition system which uses the Viola-Jones.

Keywords: Image processing,Face Autentication,Face Recognition,viola-Jonas Algorithm,Impersonation Detection,open CV

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Object Detection using Haar feature-based cascade classifiers is an effective method proposed by Paul Viola and Michael Jones in the 2001 paper, "Rapid Object Detection using a Boosted Cascade of Simple Features". It is a machine learning based approach in which a cascade function is trained from a lot of positive and negative images. It is then used to detect objects in other images.For face detection Initially, the algorithm needs a lot of positive images (images of faces) and negative images (images without faces) to train the classifier. Then we need to extract features from it. For this, Haar features are used. They are just like our convolutional kernel. Each feature is a single value obtained by subtracting the sum of pixels under the white rectangle from the sum of pixels under the black rectangle.Now all possible sizes and locations of each kernel are used to calculate plenty of features. For each feature calculation, we need to find the sum of the pixels under the white and black rectangles. To solve this, they introduced the integral images. It simplifies calculation of the sum of the pixels, how large may be the number of pixels, to an operation involving just four pixels.

But among all these features we calculated, most of them are irrelevant. For example, consider an image, Suppose Top row shows two good features. The first feature selected seems to focus on the property that the region of the eyes is often darker than the region of the nose and cheeks. The second feature selected relies on the property that the eyes are darker than the bridge of the nose. But the same windows applying on cheeks or any other place is irrelevant.To select the best features out of 160000+ features,It is achieved by Adaboost.For

this, we apply each and every feature on all the training images. For each feature, it finds the best threshold which will classify the faces to positive and negative. But obviously, there will be errors or misclassifications. We select the features with minimum error rate, which means they are the features that best classifies the face and non-face images. (The process is not as simple as this. Each image is given an equal weight in the beginning. After each classification, weights of misclassified images are increased. Then again same process is done. New error rates are calculated. Also new weights. The process is continued until required accuracy or error rate is achieved or required number of features are found).Final classifier is a weighted sum of these weak classifiers. It is called weak because it alone can't classify the image, but together with others forms a strong classifier. The paper says even 200 features provide detection with 95 percent accuracy. Their final setup had around 6000 features. (Imagine a reduction from 160000+ features to 6000 features.In an image, most of the image region is non-face region. So it is a better idea to have a simple method to check if a window is not a face region. If it is not, discard it in a single shot. Don't process it again. Instead focus on region where there can be a face. This way, we can find more time to check a possible face region.For this they introduced the concept of Cascade of Classifiers. Instead of applying all the 6000 features on a window, group the features into different stages of classifiers and apply one-by-one. (Normally first few stages will contain very less number of features). If a window fails the first stage, discard it. We don't consider remaining features on it. If it passes, apply the second stage of features and continue the process. The window which passes all stages is a face region. OpenCV already contains many pre-trained classifiers for face, eyes, smile etc. Those XML files are stored in opencv/data/haarcascades/ fold.

1.1.1 Motivation

A threat is any factor that poses any form of harm to the proper functioning and satisfying the objective of the system in a secure way ensuring confidentiality, authenticity and other necessary features. An Online Examination is a critical asset in the fast growing online learning environment. In order to assess the threats we should understand the nature of the system and must analyze the environment where the system is deployed. Collusion is identified as the highest rated threat in an online examination At present the examination system is majorly manual. The candidate appearing for the online examination is authenticated by manual verification of their credentials by the examiner. A staff manually evaluates the candidates answers and manually grades him/her. The data are manually added to the database in which errors may occur. The system does not perform any check on or before taking an examination. This lack of presence of an auto proctored examination has led to a rise of collusion(impersonation). In this project i try to design a system that to overcome this disadvantage. Facial recognition with Viola-janas algorithm is a simple approach that can embedded to the existing system to conduct dynamic authentication of the candidate. Candidate registration photograph is used for comparison with the picture recorded while the candidate enters the examination hall for identification of violations. Violation here refers to mismatch of features above a certain degree which drives to the decision that the person currently taking up the examination is not the corresponding candidate.

1.2 Objective

The system first requires registration of the candidate before the examination. On the beginning of the examination , the system takes your face and validates it and on verification, the candidate will be taken to the examination portal , he or she can download question paper and upload the answer scripts. The user's face will be captured and stored in the database so that impersonation won't happen. Impersonation is a significant problem in online examinations. An efficient invigilation mechanism is the need of the hour to ensure the standard of examination and to maintain the authentic conduct of the examination. Hence, we propose dynamic face authentication using the Viola-Jones algorithm and SVM to check the integrity of the candidate in the beginning of the examination. After downloading question paper ,student can

upload answer scripts

1.3 Contribution

The major contributions in this project are:

1. Designed and developed a new system for detecting impersonation in online examinations using Image Processing.

2.Proper security measures are introduced in the system.

3.Proper notification mechanisms are provided to the users, so that they can know the details of examinations.

4.The system is developed as a mobile application which provides any where any time access.

5.By providing identity verification at the beginning of examination automatically, it has reduced effort on human factor.

6.Providing a sophisticated authentication process.

7.This system can also be applied to other examination systems to avoid impersonation in any examination. The ease and less expensive nature of the system makes it an affordable system in all examination systems.

1.4 Report Organization

The project report is divided into five sections. Section 2 describes the literature survey that is the current scenario. Section 3 describes the methodology used for implementing the project. In methodology, workflow of the project, and sprints details are described.Section 4 gives the resultes and discussions about the project and finally section 5 gives the conclusion.

Chapter 2

Literature Survey

An Online Examination is a critical asset in the fast growing online learning environment. In order to assess the threats we should understand the nature of the system and must analyze the environment where the system is deployed. At present the examination system is majorly manual. The candidate appearing for the online examination is authenticated by manual verification of their credentials by the examiner. A staff manually evaluates the candidates answers and manually grades him/her. The data are manually added to the database in which errors may occur. The system does not perform any check on or before taking an examination. This lack of presence of an auto proctored examination has led to a rise of collusion(impersonation) .Online Examination faces many security threats. The threats are of two types, intrusive and non-intrusive. Non-intrusive is the prominent threat faced in remote online examinations. Collusion is a type of nonintrusive threat, where a student invites a third party for impersonation or assistance during the examination. Collusion is comparatively difficult to detect and prevent because of the lack of efficiency and integrity on the part of the external invigilator. So an efficient invigilation mechanism is the need of the hour to ensure the standard of examination and to maintain authentic conduct of examinationExisting system has an efficient identity check when the candidate appears for the examination by manual verification of credentials, biometrics, and image matching by the corresponding invigilator. But the system does not perform any check during the course of examinationThis lack of presence of a invigilating mechanism has led to increase of collusion. Lack of integrity of the invigilator also contributes to collusion.Currently authentication of the user before taking up the examination is carried out using biometric authentication and manualverification of credentials by the invigilator. This process

ensures authentication at entry level. As we already discussed lack of integrity of the invigilator may increase the chance of collusion in an examination. The only way to address this issue is to conduct a runtime dynamic authentication during the online Hence,this project propose dynamic face authentication using the Viola-Jones algorithm and SVM to check the integrity of the candidate in the beginning of the examination.

There are lots of research papers based on these topics like ”A New System for Face Detection based on Eigen Face and Bat Algorithm” By Nima ABEROMAND1*, Rama Haghani PANAH2, Seyed Mahdi JAMEII3 ,Bashir Bagheri NAKHJAVANLO and ” Active Testing for Face Detection andLocalization ”Raphael SznitmanDepartment of Computer Science Johns Hopkins University Bruno Jedynak Department of Applied Mathematics and Statistics Johns Hopkins University..etc.Even though many algorithms are more accurate than Haar cascades (HOG + Linear SVM, SSDs, Faster R-CNN, YOLO, to name a few), but they are still relevant and useful today. One of the primary benefits of Haar cascades is that they are just so fast — it’s hard to beat their speed.so by using haar cascade algorithm there are advantages like ,An important part of the computer vision and image processing literature Still used with OpenCV, Still useful, particularly when working in resource-constrained devices when we cannot afford to use more computationally expensive object detectors.

Chapter 3

Methodology

3.1 Introduction

We are living in the era where office work is becoming work from home and examinations are becoming online examinations. In online examinations there is a lot of chance of impersonation. This paper proposes a method to overcome the downside of online examinations which is impersonation. This approach aims in increasing the credibility of online exams and eliminate the need for an invigilator so that the exams can be taken from any convenient location. In recent days, the candidate appearing for an online examination is authenticated by carrying out manual verification of the candidate's credentials by the invigilator. Conducting an automated face authentication dynamically during the exam at definite intervals will check the identity of the user. For this type of authentication, we use facial recognition system which uses the Viola-Jones algorithm, and SVM (SUPPORT VECTOR MACHINE) for detection and recognition respectively.

3.2 Modules

The project is divided into 2 functional modules.They are ,

1. Admin

• Login - Admin login to the website for managing the students and to schedule exams.

• Student registration -Student details and their photos are registered by admin .

• Adding subjects -Admin can add subjects according to the courses of registered students

• Exam scheduling -Admin schedule exams for each courses with specified time and date.

• Adding questions and answers -Admin also provide question papers.

• Adding study materials -Admin can provide study materials to the registered students

2. Student

• Login -Registered students can login to the mobile application .

• View study materials -Students can view and download study materials .

• View exam notifications -Students get notifications of scheduled exams.

• Attend exams -Student's faces identified and verified.If the student is a registerd one he or she is allowed to access and download the question paper.if he or she is an invalid user, cannot access the question paper.After writing the question paper students can upload their answer scripts.

3.3 Developing Environment

• OPERATING SYSTEM: Windows 10 and above.

• FRONT END: Html, Css, Javacript

• BACK END: Mysql

• SOFTWARES USED: Jetbrains Pycharm, Android Studio

• TECHNOLOGY USED: Python,Java

• FRAME WORK USED: Flask

3.4 Work Flow

IDENTITY VERIFICATION

The identity of the candidate is verified by the face authentication system. The identity verification is done every 5 minutes to check if no impersonation has happened. This face authentication system uses violajones algorithm to locate or identify a face and SVM algorithm to classify the faces in the database.

A. Viola- Jones Algorithm

The Viola-Jones algorithm is a widely used algorithm for object detector. The detector shows its vigorous power in face detection upon 2 stages. They are:

1) Training

2) Detection

We have 2 steps for training phase:

1) Training

a) Training the classifiers

This part trains the system to identify the features of an image. For this identification, we're providing the system with the data, and subsequently training it to learn from the data provided.

b) Training the Adaboost

It requires plenty of facial image data to optically discern more features in different forms. This algorithm learns from the images we supply it and is able to determine the positives and negatives in the data, letting it to be more precise.

2) Detection

The Viola-Jones algorithm is specially outlined for frontal faces rather than the face looking sideways, upwards, or downwards. Afore detection, the image is first converted into a grayscale image which is the obligatory step. It first detects the face on the grayscale image, and then finds the position on the colored image. We have 2 steps in the detection phase: a) Detecting the haar- like features

There are 3 types of Haar-like features

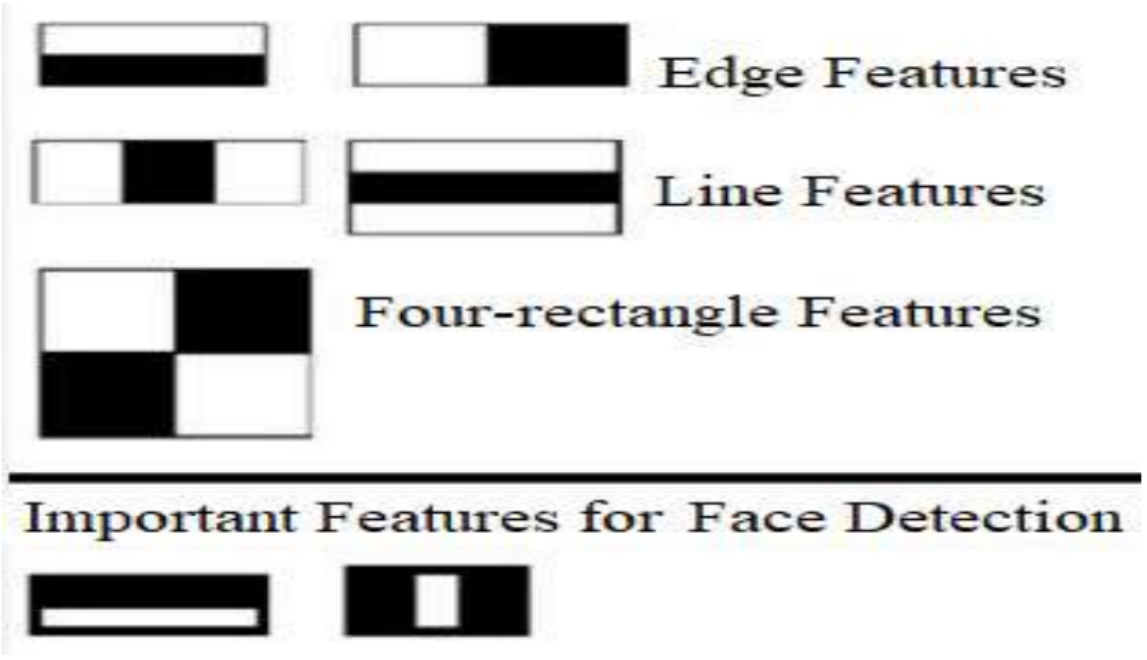

Fig 2: Haar-like features

i. Edge features

ii. Line-features, and

iii. Four-sided features

b) Creating integral images

Creating an integral image from the regular image plays its part in allowing us to perform rigorous calculations expeditiously so we can understand whether a feature of a number of features fits the criteria.

Haar Cascade Detection is one of the oldest yet powerful face detection algorithms invented. It has been there since long, long before Deep Learning became famous. Haar Features were not only used to detect faces, but also for eyes, lips, license number plates etc.. Here we slid a small matrix across our image from left-to-right and top-to-bottom, computing an output value for each center pixel of the kernel. we slid a small matrix across our image from left-to-right and top-to-bottom, computing an output value for each center pixel of the kernel. Sliding a fixed size window across the image at multiple scales. At each of these phases, the window stops, computes some features, and then classifies the region as Yes, this region does contain a face, or No, this region does not contain a face. This requires a bit of

machine learning. We need a classifier that is trained in using positive and negative samples of a face: Positive data points are examples of regions containing a face Negative data points are examples of regions that do not contain a face. For this OpenCV used to perform face detection out-of-the-box using a pre-trained Haar cascade. OpenCV ensures that we do not need to provide our own positive and negative samples, train our own classifier, or worry about getting the parameters tuned exactly right. Instead, we load the pre-trained classifier(OpenCV) and detect faces in images.

Step 1: Loop over frames from video stream or image

Step2: Read the next frame

Step3: Resize it

Step4: Convert it to grayscale.

Step5: Once the frame has been converted to grayscale, apply the face detector Haar cascade to locate any faces in the input frame

B. Svm

A support vector machines is a supervised learning algorithm. Support vector machine algorithms are used in many applications like image and speech recognition, computer vision and NLP. SVM algorithm plots each item with a value in the graph with n coordinates each coordinate representing a feature. Then the data items are classified according to their values and a hyper-plane is used to separate the classes.

The features from images associated with each person are gathered.Then a complete set of information from all of the stored images, isolated per person as a single svm label, is trained to generate a svm model.Svm classifier is used for obtaining better accuracy.

3.5 User Story

The project was developed using Agile methodology.The project has two users .First one is admin and second one is student as user.The user story of system is given in Table 3.1

User Story ID	As a <type of user>	I want to <perform some task>	So that I can <achieve some goal>
1	Admin	Login to admin account	Access the admin account
2	Admin	student registration management	Add and manage student
3	Admin	Add and manage questions and study materials	Manage questions and study materials
4	Admin	Add and manage subjects	Manage students
5	Admin	Schedule exam	Provide examination portal
6	User	Login to user account	Access the user account
7	User	View study materials	View study materials in android
8	User	View exam notifications	View notifications
9	User	Attend exams	Identity verification and face recognition

Figure 3.1: user story

3.6 Product Backlog

A product backlog is a list of the new features, changes to existing features, bug fixes, infrastructure changes or other activities that a team may deliver in order to achieve a specific outcome.The product backlog of the system is given in Table 3.2

User story ID	Priority <High/Medium/Low>	Size (Hours)	Sprint <#>	Status <Planned/In progress/Completed>	Release Date	Release Goal
1	Medium	2	1	Completed	8/01/2022	Table Design
2	High	3		Completed	8/01/2022	Form Design
3	High	5		Completed	8/01/2022	Basic coding
4	High	5	2	Completed	22/01/2022	Face Recognition
5	Medium	4		Completed	22/01/2022	Implementing SVM algorithm
6	High	10	3	Completed	5/02/2022	Creation of Online Examination Portal
7	Medium	5	4	Completed	20/02/2022	Testing Data
8	High	10		Completed	20/02/2022	Output generation

Figure 3.2: Product Backlog

3.7 Project Plan

A project plan that has a series of tasks laid out for the entire project, listing task durations, responsibility assignments, and dependencies. The Project plan is given in Table 3.3

User Story ID	Task Name	Start Date	End Date	Hours	Status
1	Sprint 1	26/12/2021	28/12/2021	10	Completed
2		29/12/2021	31/12/2021		Completed
3		3/01/2022	8/01/2022		Completed
4	Sprint 2	9/01/2021	16/01/2022	13	Completed
5		18/01/2021	22/01/2022		Completed
6	Sprint 3	23/01/2022	27/01/2022	12	Completed
7		30/01/2022	5/02/2022		Completed
8	Sprint 4	6/02/2022	10/02/2022	9	Completed
9		16/02/2022	20/02/2022		Completed

Figure 3.3: Project Plan

The Project has four sprints:

1. Sprint 1 :

In the first sprint table, form and its basic coding is completed.

2. Sprint 2 : In the second sprint Face Recognition and SVM algorithm to Classify the Faces in the Database is planned to complete.

3. Sprint 3 :

Development of Online Examination Portal planned to complete in the third sprint.

4. Sprint 4 :

In the fourth sprint planned to complete the testing and output generation.

3.8 Sprint Backlog Plan

Backlog Item	Status And Completion Date	Original Estimation in Hours	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	Day 8	Day 9	Day 10	Day 11	Day 12	Day 13	Day 14
UserStory#1,#2,#3			hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs
Table Designing	28/12/2021	2	1	1	0	0	0	0	0	0	0	0	0	0	0	0
Form Designing	31/12/2021	3	0	0	1	1	1	0	0	0	0	0	0	0	0	0
Coding	8/01/2022	5	0	0	0	0	0	1	1	1	1	1	0	0	0	0
UserStory#4, #5																
Face Recognition	16/01/2022	5	1	1	0	1	1	1	0	0	0	0	0	0	0	0
Identity Verification	22/01/2022	5	0	0	0	0	0	0	0	1	1	0	1	1	1	0
UserStory#6,#7																
Online Examination Portal	27/01/2022	5	1	1	1	0	1	1	0	0	0	0	0	0	0	0
UserStory#8,#9																
Testing Data	10/02/2022	10	1	1	1	1	1	0	0	0	0	0	0	0	0	0
Output Generation	20/02/2022	9	0	0	2	1	1	0	2	2	1	0	0	0	0	0
Total		44	4	4	5	4	5	3	3	5	4	2	2	2	1	0

Figure 3.4: Sprint BackLog

3.9 Sprint Actual

Backlog Item	Status And Completion Date	Original Estimation in Hours	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	Day 8	Day 9	Day 10	Day 11	Day 12	Day 13	Day 14
UserStory#1,#2,#3			hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs
Table Designing	28/12/2021	2	1	1	0	0	0	0	0	0	0	0	0	0	0	0
Form Designing	31/12/2021	4	0	0	0	1	1	0	1	1	0	0	0	0	0	0
Coding	8/01/2022	4	0	0	0	0	0	0	0	0	2	1	1	0	0	0
UserStory#4, #5																
Face Recognition	16/01/2022	10	2	1	1	0	1	1	1	0	1	1	0	1	0	0
Identity Verification	25/01/2022	10	1	1	0	1	1	0	0	1	2	0	0	1	1	1
UserStory#6,#7																
Online Examination Portal	27/01/2022	5	1	0	0	0	1	1	1	0	0	0	1	0	0	0
UserStory#8,#9																
Testing Data	11/02/2022	10	1	1	1	0	0	1	1	1	0	0	1	1	1	1
Output Generation	20/02/2022	5	0	1	1	1	1	1	0	0	0	0	0	0	0	0
Total		50	6	5	3	3	5	4	4	3	5	2	3	3	2	2

Figure 3.5: Sprint Actual

Chapter 4

Results and Discussions

4.1 Results

To perform Face Authentication we use image processing toolbox to perform the face authentication. Just like any other form of biometric identification, face recognition requires samples to be collected, identified, extracted with necessary (features) information, and stored for recognition. The algorithm that is used for face recognition is Viola Jones algorithm and SVM (Support Vector Machine).Dlib and openCV library with python is used for the face recognition algorithm and for image processing. After identifying the face, next step is feature extraction. HOG(Histogram of Oriented Gradients) features are extracted from the image and then start comparing with the image that is uploaded during the registartion of student by admin stored in the database to identify whether he/she is a valid user or not. If it matches up to 60 percent to 70 percent he/she is a registered user. If it is not satisfying the threshold value.That is, it matches only up to 20 percent to 30 percent then he/she is an invalid user. After face authentication, the candidate is provided with the examination portal where he/she can download the question paper and upload their answer scripts within the alloted time.

Chapter 5

Conclusions

Online Exam is a moblie application.It is easy to use and more interactive to the users. At present the examination system is majorly manual.The candidate appearing for the online examination is authenticated by manual verification of their credentials by the examiner.The system does not perform any check on or before taking an examination.This lack of presence of an auto proctored examination has led to a rise of collusion(impersonation).As a solution in this project, i developed a system which help an examiner to conduct a fully automated examination system where the user can take the exam from anywhere and the authenticity of the candidate. The assessment is auto proctored and so there is no need for an examiner. The user's face is verified dynamically to check is identity at the beginning of examination.

The advantages of employing facial recognition system to detect impersonation are : Easy to implement,Sophisticated authentication process, Higher accuracy, Affordable, Reduced effort on human factor, Reliability.This concept can be applied not only in remote online examinations but also can be applied to all other online examinations to prevent these threats.This system can also be applied to other examination systems to avoid impersonation in any examination. The ease and less expensive nature of the system makes it an affordable system in all examination systems. The project is developed using Android as front end and MySQL as back end. The project has been developed, tested , documented and implemented successfully.This has been developed as versatile and user friendly as possible. At the final stage of this project with a proud feeling that some thing new had developed.

References

[1] **Raphael Sznitman and Bruno Jedynak(** (2010) “Active Testing for Face Detection And Localization”

[2] **Rajeev Ranjan, Vishal M. Patel and Rama Chellappa** (2017)“HyperFace: A Deep Multi-Task Learning Framework for Face Detection, Landmark Localization, Pose Estimation, and Gender Recognition”

[3] **Shengcai Liao, Anil K. Jain and Stan Z. Li.** (2016) “A Fast and Accurate Unconstrained Face Detector”.

[4] **Zulhadi Zakaria and Shahrel A. Suandi** (2012) “Face detection using combination of Neural Network and Adaboost”.

[5] **Zulhadi Zakaria and Shahrel A. Suandi** (2012) “Face detection using combination of Neural Network and Adaboost”.

[6] **Lutz Goldmann, Ullrich J. Monich and Thomas Sikora** (2007) “Components and Their Topology for Robust Face Detection in the Presence of Partial Occlusions”

Appendix

Source Code

```
# USAGE
# python encode_faces.py --dataset dataset --encodings encodings.pickle

# import the necessary packages
# from imutils import paths
import face_recognition
import argparse
import pickle
import cv2
import os

# # construct the argument parser and parse the arguments
# ap = argparse.ArgumentParser()
# ap.add_argument("-i", "--dataset", required=True,
#  help="path to input directory of faces + images")
# ap.add_argument("-e", "--encodings", required=True,
#  help="path to serialized db of facial encodings")
# ap.add_argument("-d", "--detection-method", type=str, default="cnn",
#  help="face detection model to use: either `hog` or `cnn`")
# args = vars(ap.parse_args())
def enf(fn):
# grab the paths to the input images in our dataset
  print("[INFO] quantifying faces...")
  imagePaths = [fn]

  # initialize the list of known encodings and known names
  knownEncodings = []
  knownNames = []

  # loop over the image paths
  for (i, imagePath) in enumerate(imagePaths):
    # extract the person name from the image path
    print("[INFO] processing image {}/{}".format(i + 1,
      len(imagePaths)))
    print("imagepath-------",imagePath)
    name = "face"
    print("id=",name)
    # load the input image and convert it from RGB (OpenCV ordering)
    # to dlib ordering (RGB)

    image = cv2.imread(imagePath)
    rgb = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)

    # detect the (x, y)-coordnates of the bounding boxes
```

```
    # corresponding to each face in the input image
    boxes = face_recognition.face_locations(rgb,
      model='hog')

    # compute the facial embedding for the face
    encodings = face_recognition.face_encodings(rgb, boxes)

    # loop over the encodings
    for encoding in encodings:
      # add each encoding + name to our set of known names and
      # encodings
      knownEncodings.append(encoding)
      knownNames.append(name)

  # dump the facial encodings + names to disk
  print("[INFO] serializing encodings...")
  data = {"encodings": knownEncodings, "names": knownNames}
  f = open('faces.pickles', "wb")
  f.write(pickle.dumps(data))
  f.close()
# enf()


# USAGE
# python recognize_faces_image.py --encodings encodings.pickle --image examples/example_01.png

# import the necessary packages
import face_recognition

import argparse
import pickle
import cv2


def rec_face_image(imagepath):
  print(imagepath)


  idddss=imagepath.split('/')
  print(idddss)

  ap = argparse.ArgumentParser()

  data = pickle.loads(open('faces.pickles', "rb").read())

  # load the input image and convert it from BGR to RGB
  image = cv2.imread(imagepath)
  #print(image)
  h,w,ch=image.shape
  print(ch)
  rgb = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)

  # detect the (x, y)-coordinates of the bounding boxes corresponding
  # to each face in the input image, then compute the facial embeddings
  # for each face
  print("[INFO] recognizing faces...")
  boxes = face_recognition.face_locations(rgb,
    model='hog')
  encodings = face_recognition.face_encodings(rgb, boxes)

  # initialize the list of names for each face detected
  names = []
```

Appendix

```
# loop over the facial embeddings
for encoding in encodings:
  # attempt to match each face in the input image to our known
  # encodings
  matches = face_recognition.compare_faces(data["encodings"],
    encoding)
  name = "Unknown"

  # check to see if we have found a match
  if True in matches:
    # find the indexes of all matched faces then initialize a
    # dictionary to count the total number of times each face
    # was matched
    matchedIdxs = [i for (i, b) in enumerate(matches) if b]
    counts = {}

    # loop over the matched indexes and maintain a count for
    # each recognized face face
    for i in matchedIdxs:

      name = data["names"][i]
      counts[name] = counts.get(name, 0) + 1
    print(counts, " rount ")
    # determine the recognized face with the largest number of
    # votes (note: in the event of an unlikely tie Python will
    # select first entry in the dictionary)

    name = max(counts, key=counts.get)


  if name!="Unknown":
    names.append(name)

  print(names,"names are")
return names
```

Database Design

Attribute Name	Datatype	Length	Description
ID	Int	11	Primary Key
User Name	Varchar	40	Unique
Password	Varchar	40	unique
Type	Varchar	20	

Table A.1: Login

Attribute Name	Datatype	Length	Description
ID	Int	11	Primary Key
Login id	Int	11	
First name	Varchar	20	
Middle name	Varchar	20	
Last name	Varchar	20	
DOB	Varcharr	20	
Gender	Varchar	20	
Course	Int	20	
Place	Varchar	10	
Post	Varchar	10	
Pin	bigint	20	
Phone	bigint	20	
Email	Varchar	20	
Date of admission	Varchar	20	

Table A.2: Student

Attribute Name	Datatype	Length	Description
SID	Int	11	Primary Key
Subject	Varchar	40	
Course	Varchar	40	
Description	Varchar	100	

Table A.3: Subject

Attribute Name	Datatype	Length	Description
Ans id	Int	10	Primary Key
Student id	Int	10	
Ans paper	text	10	
Exam id	Int		
Date	date		
Time	time		

Table A.4: Answer

Attribute Name	Datatype	Length	Description
id	Int	5	Primary Key
Course	Varchar	20	
Description	Varchar	40	

Table A.5: Course

Attribute Name	Datatype	Length	Description
id	Int	11	Primary Key
sid	Int	11	
Date	Varchar	20	
Time	Varchar	50	
Q paper	Varchar	100	
Duration	Varchar	20	

Table A.6: Exam

Attribute Name	Datatype	Length	Description
id	Int	5	Primary Key
Stu id	Varchar	20	
Pics	Varchar	40	

Table A.7: Pics

Attribute Name	Datatype	Length	Description
id	Int	11	Primary Key
Sub id	Intr	11	
material	Varchar	300	

Table A.8: Study material

DaTaflow Diagram

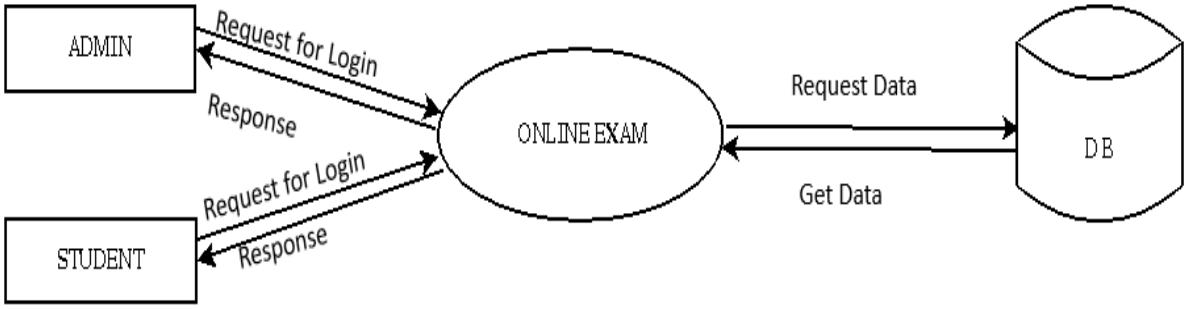

Figure A.1: LEVEL 0

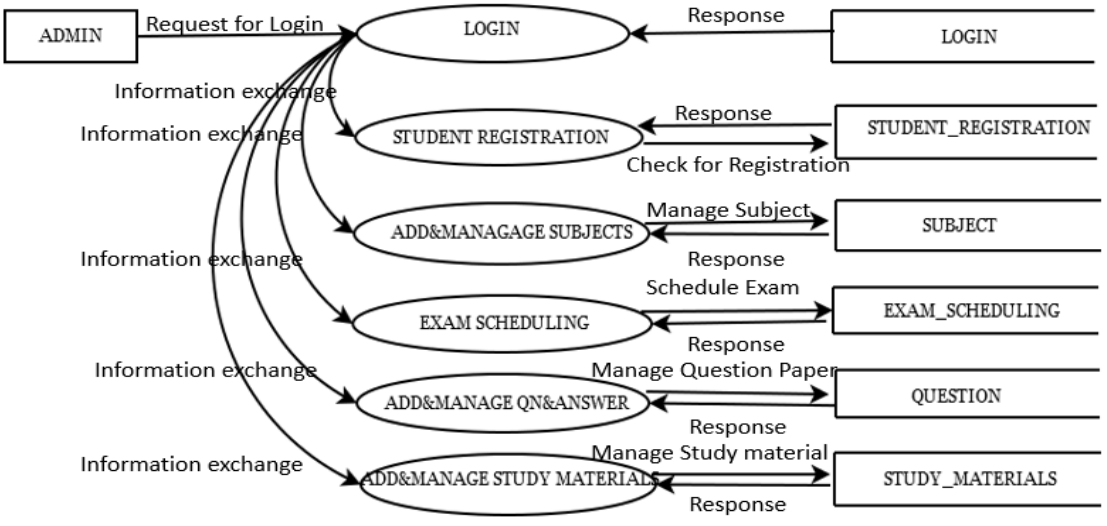

Figure A.2: LEVEL 1.1

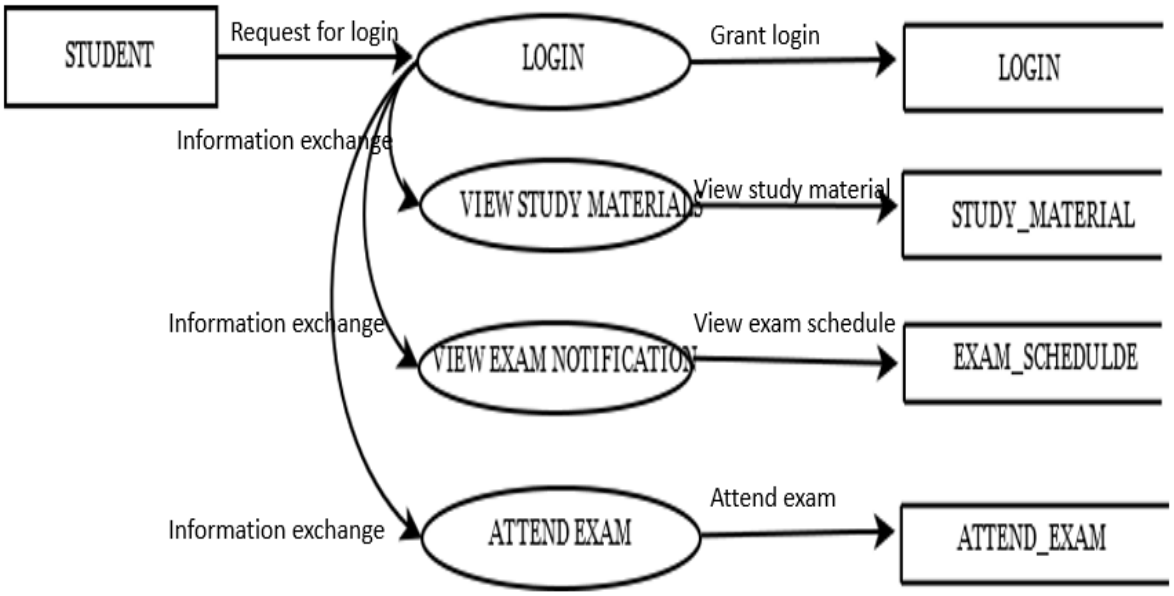

Figure A.3: LEVEL 1.2

User Interface

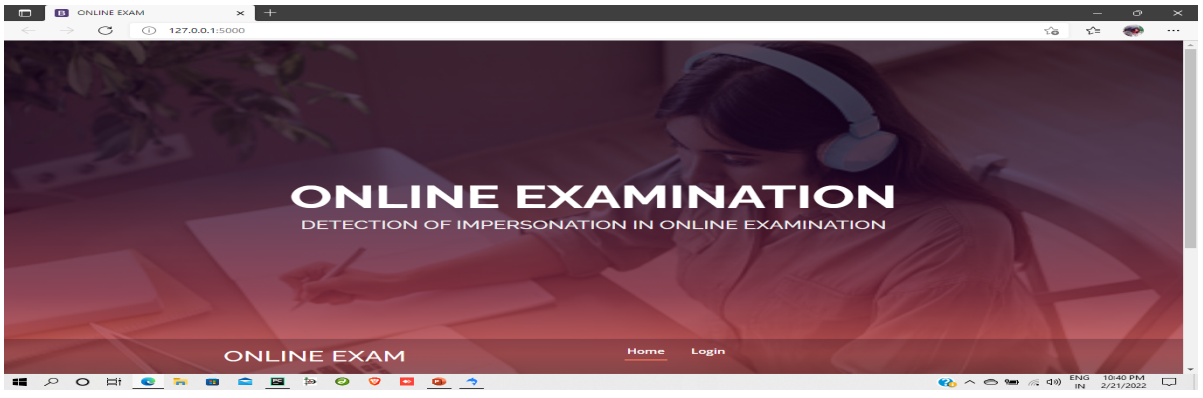

Figure A.4: Home Page

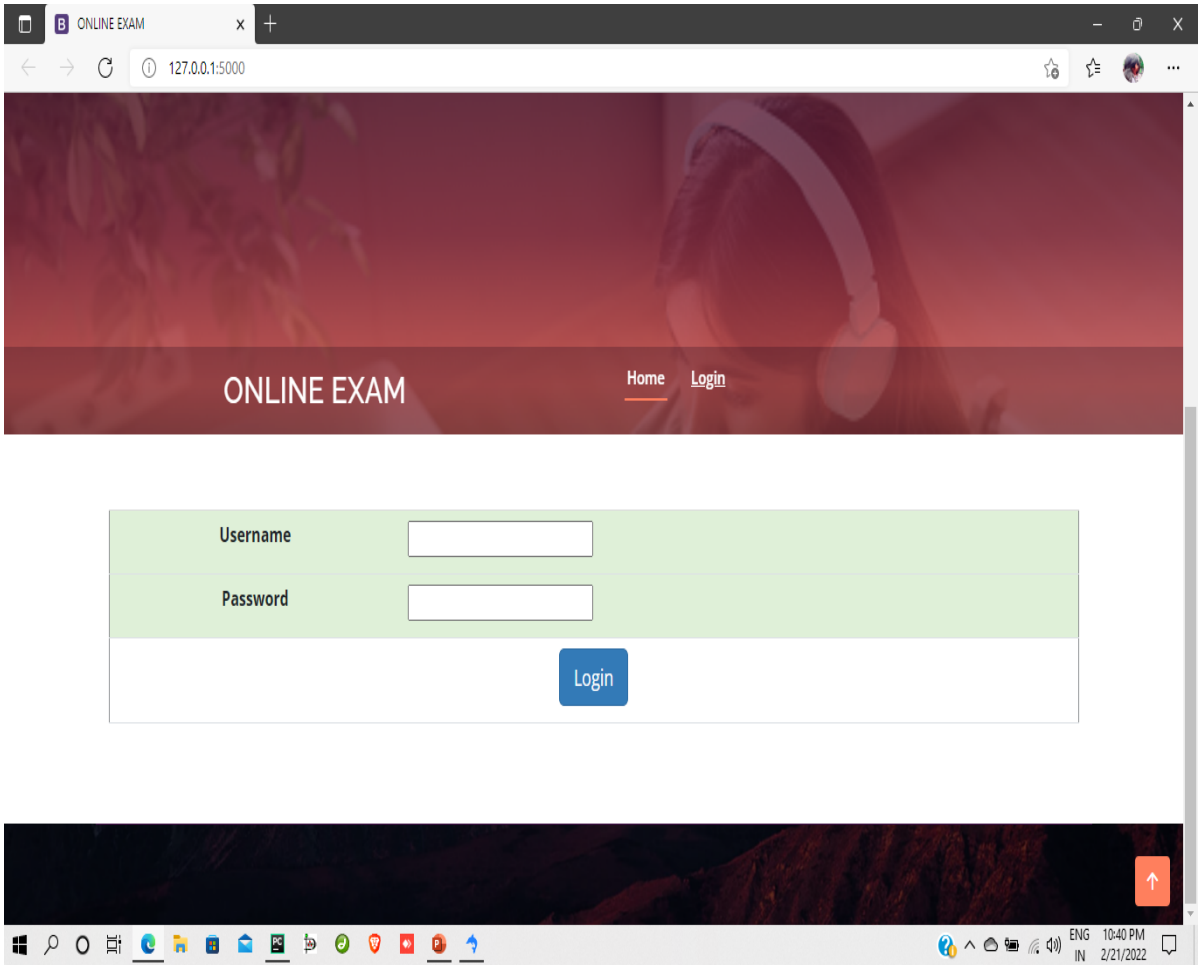

Figure A.5: Login Page

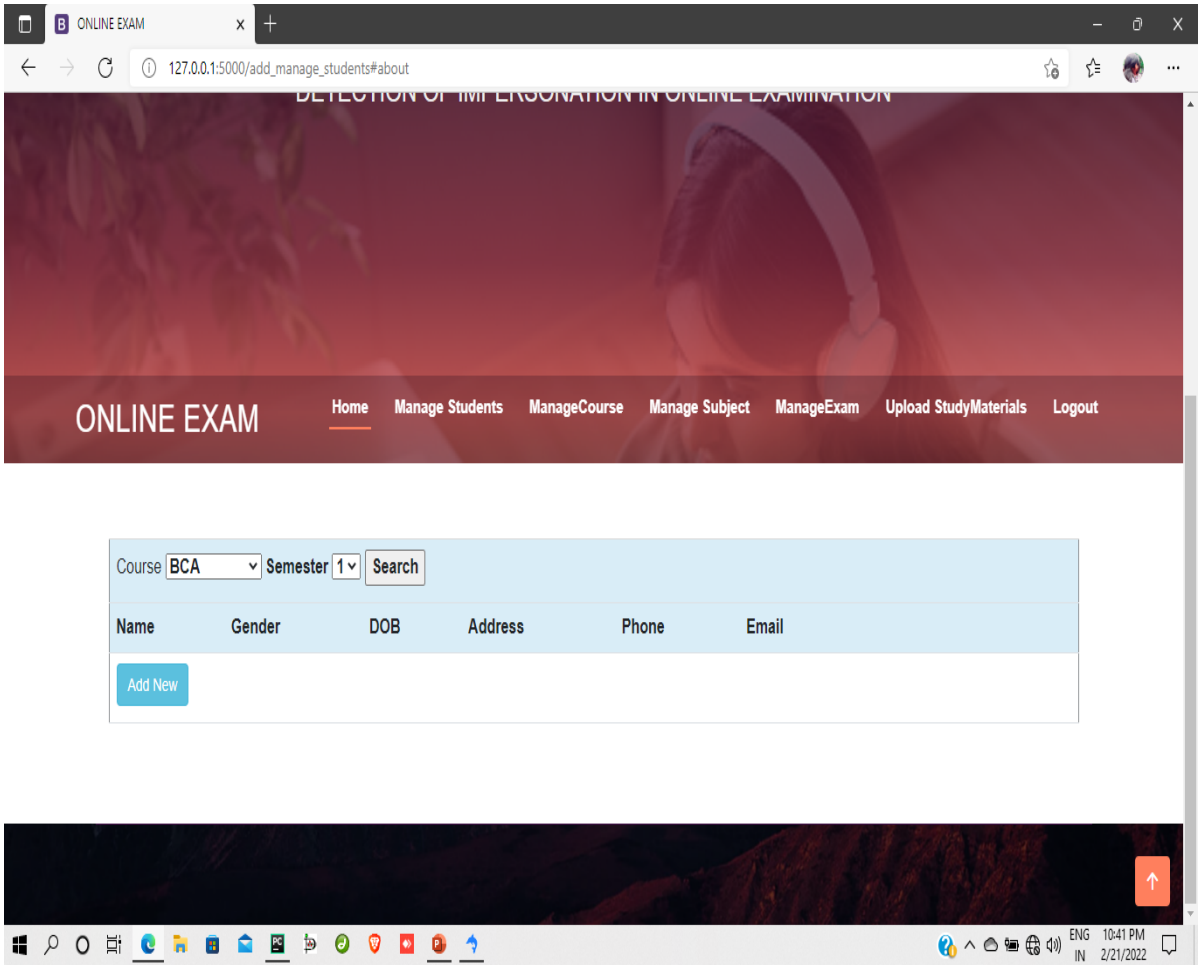

Figure A.6: Manage Students

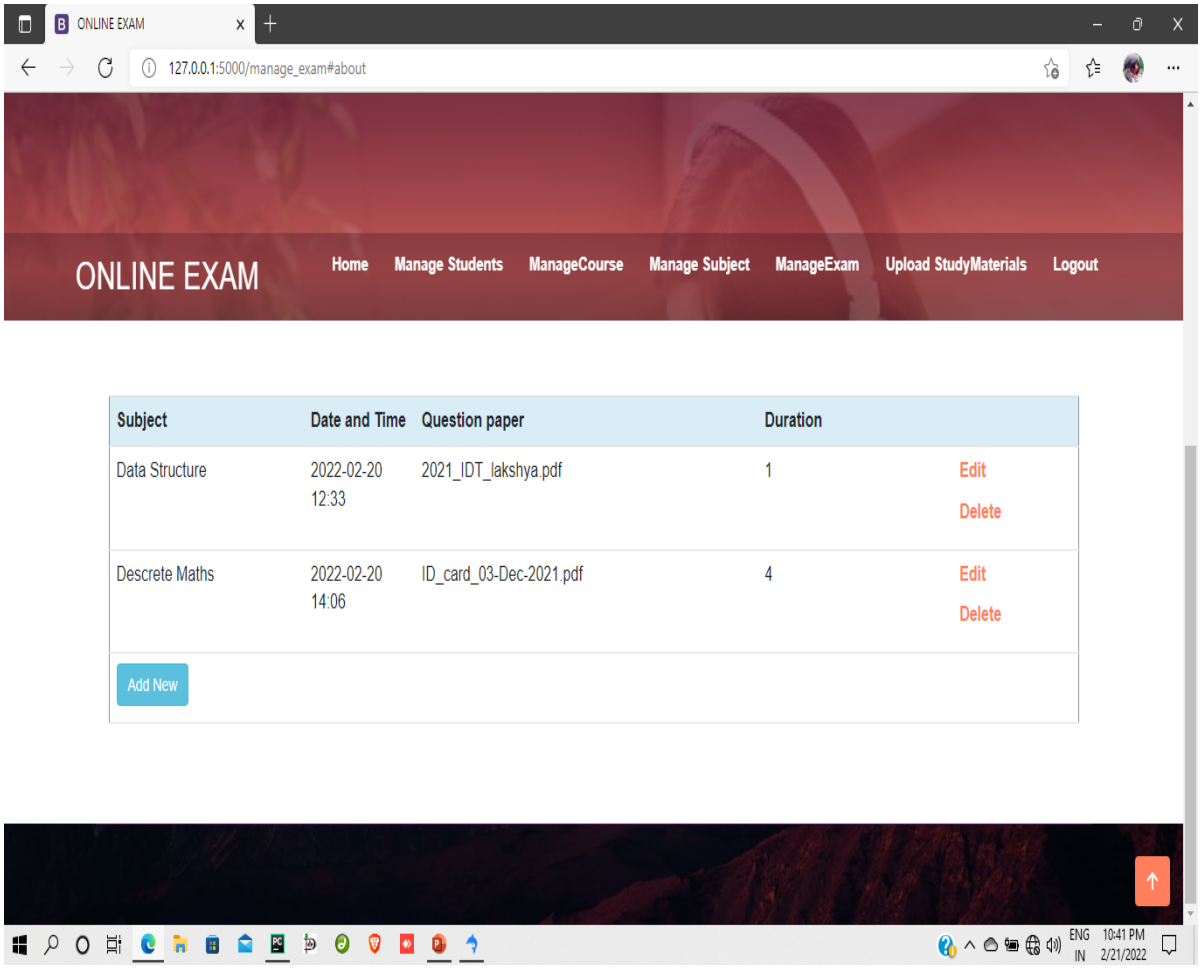

Figure A.7: Manage Exam

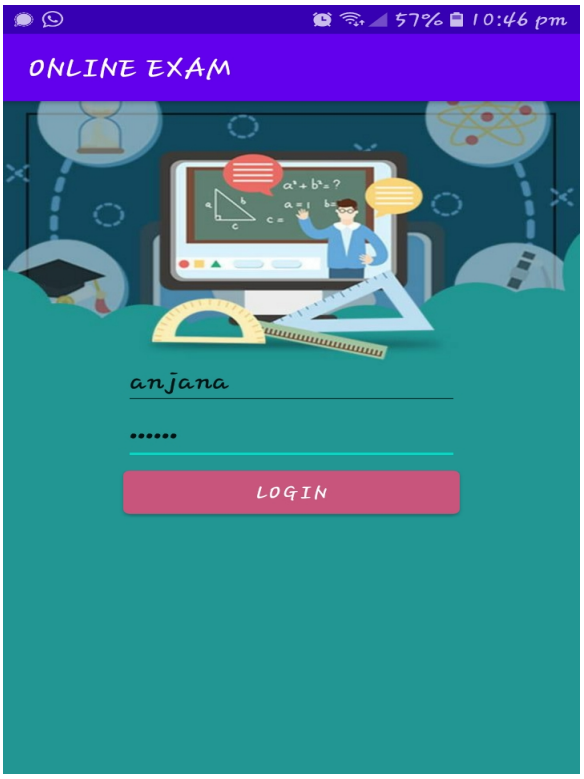

Figure A.8: Login

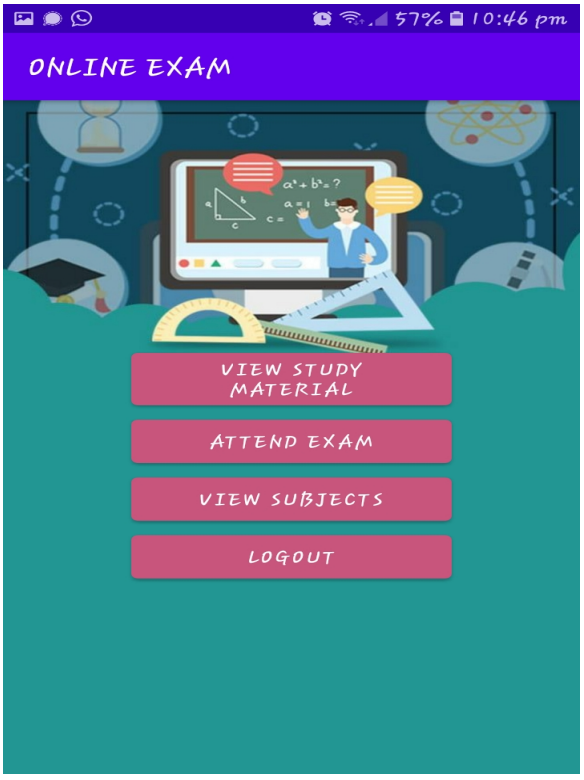

Figure A.9: Home

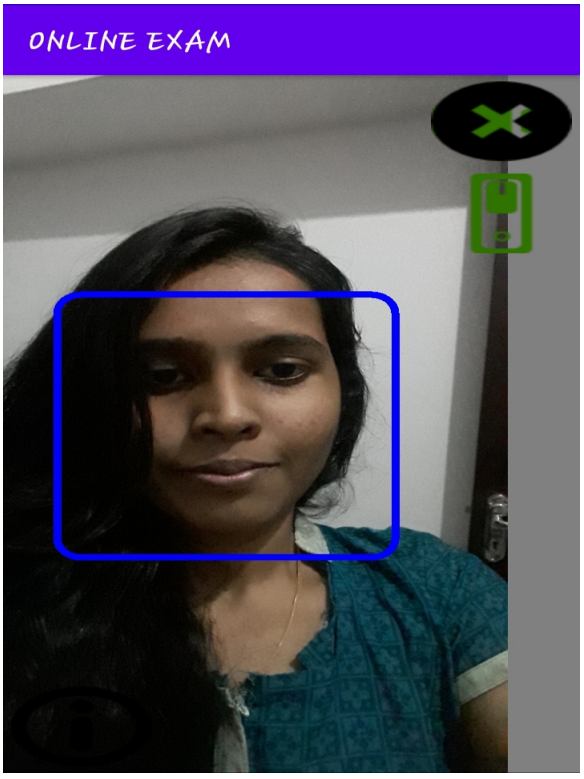

Figure A.10: Face Recognition